PES UNIVERSITY, Bengaluru
Department of Computer Science and Engineering
B. Tech (CSE) – 5th Semester – Aug-Dec 2023

UE21CS341A - Software Engineering

SOFTWARE REQUIREMENTS SPECIFICATION

PROJECT TITLE

VERSION 1

Team #: T01

PES1UG21CS578	Shreya Sridhar
PES1UG21CS584	Shriansh Mohanty

PES1UG21CS589	Shubha Masti
PES1UG21CS935	Shyam Krishna Sateesh

Organization : PES University
Date Created : 05/10/2023

Table of Contents

Table of Contents 1

Revision History

1. Introduction

1.1 Purpose 4

1.2 Intended Audience and Reading Suggestions 4

1.3 Product Scope 5

1.4 References 6

2. Overall Description

2.1 Product Perspective 7

2.2 Product Functions 7

2.3 User Classes and Characteristics 8

2.4 Operating Environment 10

2.5 Design and Implementation Constraints 10

2.6 Assumptions and Dependencies 11

3. External Interface Requirements

3.1 User Interfaces 12

3.2 Software Interfaces 14

3.3 Communications Interfaces 15

4. Analysis Models *16*

5. System Features

5.1 User Registration/ Login 17

5.2 User Profile Management 17

5.3 Recipe Recommendation 17

5.4 Bookmark Favourites 18

5.5 Recipe Details 18

6. Other Nonfunctional Requirements

6.1 Performance Requirements 19

6.2 Safety Requirements *19*
6.3 Security Requirements *20*
6.4 Software Quality Attributes *21*
6.5 Business Rules *22*
7. Other Requirements
Appendix A: Glossary *23*
Appendix B: Field Layouts *30*
Appendix C: Requirement Traceability matrix *30*

Revision History

Name	Date	Reason for Change	Version

1. Introduction

1.1 Purpose

The primary purpose of the Recipe Recommendation System is to simplify the meal planning process for users and enhance their cooking experience. It aims to address the common dilemma of "What should I cook today?" by offering tailored culinary suggestions that align with individual needs and tastes.

This Software Requirements Specification (SRS) covers the entire Recipe Recommendation System. It includes all the functional features, data processing components, and user interfaces of the system. The scope encompasses both the frontend and backend aspects of the application, from managing user profiles and pantry ingredients to recommending recipes, as well as features related to recipe rating and history

1.2 Intended Audience

Developers: Developers will refer to this document to understand the functional requirements, technical specifications, and design considerations needed to build the system.

Project Managers: Project managers will use this document to gain a comprehensive understanding of the project's requirements, scope, and goals. It will help them plan and manage the development process.

Marketing Staff: Marketing staff may refer to this document to understand the system's features and capabilities, which can be used in promotional materials and marketing campaigns.

Users: End users of the Recipe Recommendation System will benefit from this document as it outlines the system's features and functionalities, helping them understand how to use the system effectively.

Aspiring Home Cooks: Individuals interested in expanding their culinary skills and looking for new recipes to try.

Gym Freaks: Fitness enthusiasts who wish to align their meals with their fitness goals.

Allergy-Prone Individuals: People with specific dietary restrictions or allergies who often find it challenging to locate suitable recipes.

Testers: Testers will use this document to create test cases and scenarios for validating that the system meets its requirements and functions as intended.

Documentation Writers: Those responsible for creating user manuals, help guides, and other documentation will use this SRS to ensure that the documentation accurately reflects the system's features and usage.

1.3 Product Scope

Simplify Meal Planning: The software aims to simplify the often complex and time-consuming process of meal planning. It helps users answer the common question, "What should I cook today?" by offering tailored culinary suggestions.

Personalization: By considering a user's pantry inventory, culinary preferences, dietary needs, and other factors, the system provides personalized recipe recommendations. This personalization enhances the user's cooking experience.

Efficiency and Convenience: Users can efficiently plan their meals based on the ingredients they already have at home. This not only saves time but also reduces food waste and the need for frequent grocery shopping.

Catering to Diverse Users: The system is designed to cater to a diverse range of users, from novice home cooks to health-conscious individuals with specific dietary requirements. It accommodates various culinary preferences and dietary restrictions.

Feedback and Improvement: Users have the option to rate recipes they've tried and bookmark their favourites. This feedback loop benefits both users, who can easily access their preferred recipes, and the system, which can continuously improve its recommendations.

The Recipe Recommendation System (RRS) is focused on future growth, targeting a 30% increase in daily users within its first year. We're committed to reducing food waste by 20% among our users by making sure our user base has only optimal usage of food and ingredients. Our goal is to expand our recipe database by 50% in 18 months by expanding into other cuisines, and in turn bring in more users

1.4 References

Python Documentation

Python Official Documentation

URL: https://www.python.org/doc/

React Documentation

JavaScript Library for Building User Interfaces

URL: https://reactjs.org/

MySQL Connector/Python Developer Guide

Connecting MySQL and Python

URL: https://dev.mysql.com/doc/connector-python/en/

Flask Documentation

Web Development with Python and Flask

URL: https://flask.palletsprojects.com/en/2.0.x/

pip Documentation

Package Installer for Python

URL: https://pip.pypa.io/en/stable/

GDPR Information

Regulation on Data Protection and Privacy in the European Union

URL: https://gdpr-info.eu/

Flask-Login

User Authentication with Flask-Login

URL: https://flask-login.readthedocs.io/en/latest/

Material-UI Documentation

A popular React UI framework

URL: https://mui.com/

Selenium Documentation

Web Testing Framework

URL: https://www.selenium.dev/documentation/en/

HTML 5.3 W3C Recommendation

Web Markup Language Standard

URL: https://www.w3.org/TR/2023/REC-html53-20230126/

Agile 101

What is Agile Development?

URL: https://www.agilealliance.org/agile101/

GitHub Docs

Managing Your Work on GitHub

URL: https://docs.github.com/en/issues

2. Overall Description

2.1 Product Perspective

The Recipe Recommendation System described in this SRS is a new, self-contained product developed to meet the growing demand for personalized recipe recommendations based on available pantry ingredients and user preferences. It is not a replacement for any existing systems but rather a standalone solution aimed at enhancing the culinary experiences of a diverse user base.

2.2 Product Functions

In terms of prioritization, while all user classes offer valuable insights, the Novice Home Cooks, Fitness Enthusiasts, and Users with Dietary Restrictions are most critical. They are expected to form the bulk of the user base and will most frequently rely on the system's core functionalities.

Recommend Dish

- Analyses user's input and available pantry items.
- Utilizes an algorithm to generate tailored recipe suggestions
- Returns a curated list of dishes based on preferences and ingredients.

Retrieve Past Recipes

- Accesses the user's history in the database
- Retrieves and lists previously viewed or tried recipes.
- Helps users recall and revisit their favourite or recent dishes.

List Ingredients

- Allows access to a list of all ingredients and what recipes they are used in
- Helps understanding or decision making for the end user

Browse Recipe Catalogue

- Displays the entire repository of available recipes
- Allows users to manually search, view, and select dishes.
- Organizes recipes in categories for easier navigation.

Apply Advanced Filters

- Offers specialized filtering options to users.
- Sets criteria like calorie count, allergies, or dietary preferences.
- Refines and narrows down the recipe list based on the applied filters.

2.3 User Classes and Characteristics

Novice Home Cooks

- Frequency of Use: Might use the system intermittently, particularly during weekends or special occasions.
- Functionalities Used: Primarily use basic recipe recommendations based on pantry ingredients.
- Technical Expertise: Basic to moderate. Prefer straightforward, user-friendly interfaces.
- Pertinent Characteristics: Interested in easy-to-follow recipes, may prefer video or pictorial step-by-step instructions.
- Importance: High. This group constitutes a significant portion of potential users.

Fitness Enthusiasts

- Frequency of Use: Regular users, especially around workout schedules.
- Functionalities Used: Recipe recommendations, with emphasis on health and dietary requirements.
- Technical Expertise: Moderate. Comfortable with advanced filtering and preference settings.
- Pertinent Characteristics: Focused on nutritional data, lean proteins, low carbs, etc.
- Importance: High. Their feedback can enhance the health-centric aspects of the system.

Users with Dietary Restrictions

- Frequency of Use: Regular, each time they plan a meal.
- Functionalities Used: Detailed filtering for allergies, vegan, vegetarian, gluten-free options, etc
- Technical Expertise: Moderate to high. Require detailed customization.
- Pertinent Characteristics: Need accurate and trustworthy recipe recommendations, avoiding allergens or specific ingredients.
- Importance: Crucial. Ensuring safety and satisfaction for this group is paramount.

Culinary Experts

- Frequency of Use: Occasional, looking for unique or exotic recipes.
- Functionalities Used: Advanced recipe search, might contribute or rate recipes.
- Technical Expertise: High. Might delve deep into recipe intricacies and variations.
- Pertinent Characteristics: Interested in gourmet recipes, rare ingredients, and culinary techniques.
- Importance: Moderate. Their expertise can enhance the system's credibility and diversity.

Casual Browsers

- Frequency of Use: Infrequent, might use the system on a whim or out of curiosity.
- Functionalities Used: Basic browsing and random recipe exploration.
- Technical Expertise: Basic.
- Pertinent Characteristics: Not particularly focused on cooking; more exploratory in nature.
- Importance: Low to moderate. Can offer insights into user-friendliness and general appeal.

2.4 Operating Environment

The Recipe Recommendation System is designed as a web application intended for optimal operation in the following environment:

- *Hardware Platform*: The software is platform-independent and can be accessed from any device with sufficient processing capabilities, including desktops, laptops
- *Operating System*: The system is OS-agnostic. However, for the best user experience, it's recommended to use updated versions of mainstream operating systems such as Windows, macOS, Linux
- *Web Browsers*: The software is specifically optimized for modern web browsers, with primary focus on:
    - Google Chrome (latest version recommended for best performance and compatibility).
    - Mozilla Firefox (again, latest version is recommended).
- *Other Software Components*: No specific additional software components are required. However, users should ensure that JavaScript is enabled, and cookies are allowed for personalized user experiences.
- *Network*: A stable internet connection is necessary for accessing the web app and its full range of features.
- *Coexistence*: The application is designed to peacefully coexist with other web applications and will not interfere with other browser tabs or processes. It is developed adhering to standard web protocols ensuring compatibility and smooth integration.

2.5 Design and Implementation Constraints

Language and Framework Limitations

- Python and Flask Performance dips might be observed if concurrent users exceed 10,000 in real-time.
- React Ensure 99.5% compatibility across browsers with JavaScript enabled by Q2 2024.

Dataset Collection

- Aim to gather 50,000 diverse and high-quality recipes by Q4 2023.
- Implement periodic data quality checks every 6 months.

Memory and Storage

- Pandas: Ensure backend server has at least 16GB RAM to efficiently handle large CSV files.
- Optimize data reading mechanisms by Q1 2024 to reduce memory overhead by 20%.

Integration and Compatibility

- Dedicate Q3 2023 for thorough testing and integration between Flask and React.
- Ensure 98% responsive design compatibility across devices ranging from 5-inch screens to 32-inch monitors by Q1 2024.

Deployment and Scalability

- Aim for a backend capable of scaling to handle up to 15,000 concurrent requests by Q2 2024.
- Stress-test the React frontend under load by Q3 2023 to ensure performance benchmarks are met.

Security Considerations

- Implement robust security protocols, including protection against common threats, by Q4 2023.
- Achieve HTTPS implementation across all app interfaces by Q1 2024.

Maintenance and Updates

- Schedule biannual updates for React and Flask to address potential browser and dependency changes.
- Allocate 4 weeks in Q2 and Q4 of each year for maintenance and updates.

External Libraries and Dependencies:

- Conduct quarterly reviews of all external libraries for potential updates or known issues.
- Set aside a 2-week buffer in Q3 2023 for addressing unexpected dependency- related issues.

2.6 Assumptions and Dependencies

Assumptions

- *Data Availability*: It is assumed that a sufficient and reliable dataset of recipes with detailed information, including ingredients, instructions, and nutritional data, will be available for use in the Recipe Recommendation System. The accuracy and completeness of this data could affect the system's performance.

- *Internet Connectivity*: The system assumes that users will have access to a stable internet connection to interact with the software. A lack of internet access could limit the system's functionality, especially when retrieving recipes or updates.
- *User Pantry Accuracy*: The accuracy of the ingredients users add to their digital pantry is assumed. The system relies on this information to provide recipe recommendations, and inaccurate pantry data may result in less relevant suggestions.

Dependencies

- *Third-Party Libraries and APIs*: The project may depend on third-party libraries or APIs for various functionalities, such as machine learning algorithms for recipe recommendations, nutritional data sources, or user authentication. Any changes or disruptions to these external dependencies could affect the system's functionality.
- *Database Management System (DBMS)*: The project relies on a database system for storing user data, recipes, and pantry information. The choice of the DBMS and its proper functioning are critical to the system's performance and data integrity.
- *Hardware and Hosting*: The system's deployment depends on suitable hardware and hosting infrastructure. The availability and reliability of these resources could impact system uptime and performance.
- *External Data Sources*: The system may rely on external data sources for real-time information, such as ingredient prices or availability. Changes or unavailability of these sources may affect certain features, such as cost estimation for recipes.

3. External Interface Requirements

3.1 User Interfaces

Login Page:

- Purpose: The login page allows users to access their accounts and personalize their experience.
- Components:
  i. Username and password fields.
  ii. "Remember Me" option for persistent login.
  iii. "Forgot Password" link for password recovery.
  iv. "Sign Up" link or button for new users.

v. Error Handling: Display error messages for incorrect login credentials.

Landing Page

- Purpose: The landing page provides an initial user experience, suggesting recipes based on previous history and viewed recipes.
- Components

  i. Recipe recommendations based on user preferences, and history.
  ii. Navigation to browse and search for more recipes.
  iii. Option to customize preferences.

Profile Page

- Purpose: The profile page allows users to manage their account settings and view personalized information.
- Components

  iv. User information (e.g., name, profile picture).
  v. Dietary requirements and preferences (e.g., vegetarian, gluten-free).
  vi. Preferred cuisines (e.g., Italian, Asian).
  vii. Option to edit profile information.

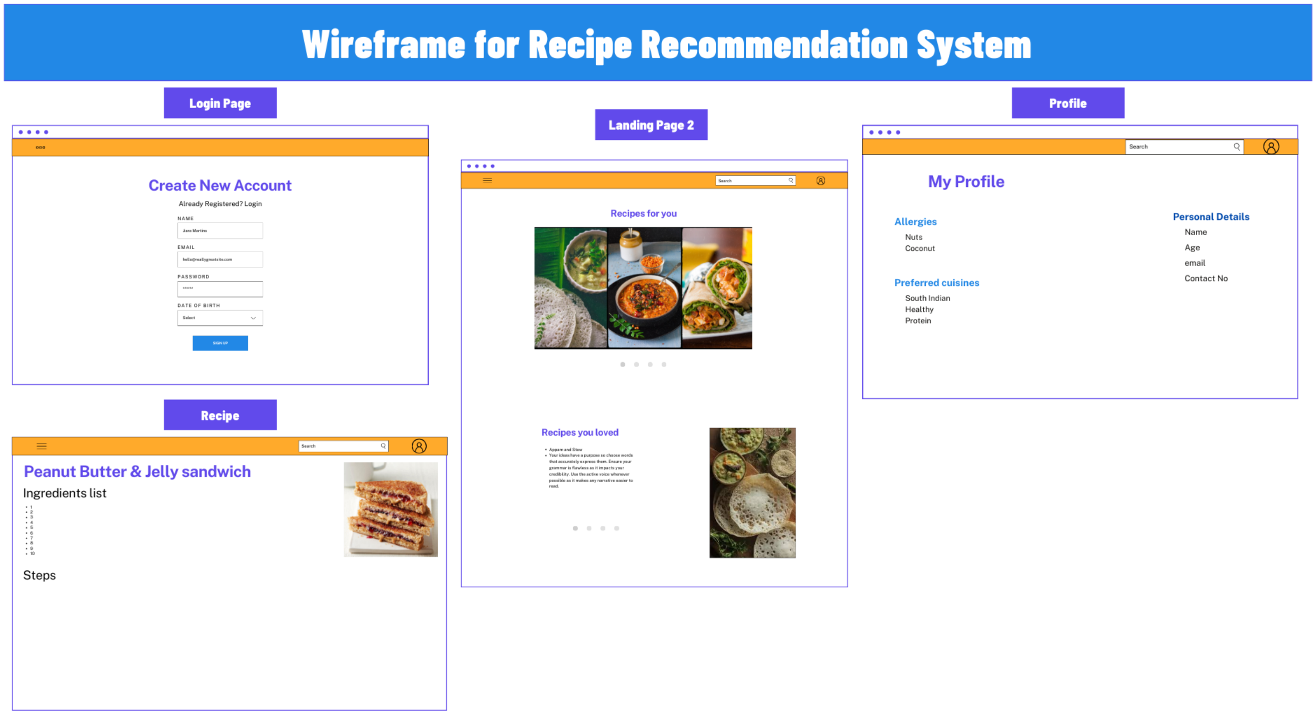
User History:

- Display a list of saved favourites.

Recipe Page:

- Purpose: The recipe page presents detailed information about a selected recipe.
- Components

  viii. Recipe name, cuisine type, and cooking time.
  ix. Ingredients list with quantities.
  x. Step-by-step cooking instructions.
  xi. Serving size and cooking tips.



- Appam and Stew
- Your ideas have a purpose so choose words that accurately express them. Ensure your grammar is flawless as it impacts your credibility. Use the active voice whenever possible as it makes any narrative easier to read.

3.2 Software Interfaces

Database: MySQL (Version: 8.0)

- Connection is established using the Python MySQL connector for data transfers.
- Incoming data includes user registration details, ingredient data, user preferences, feedback, and bookmarks.
- Outgoing data consists of recipe recommendations, user profile data, bookmarked recipes, and historical data of user interactions.

Backend Framework: Flask (Version: 2.x)

- Flask directly interfaces with the React frontend using HTTP/HTTPS requests.
- Incoming data from the frontend consists of user inputs like ingredients added to the pantry, user preferences, and feedback.
- Outgoing data sent to the frontend includes recipe recommendations, user data, and other relevant information.

Frontend Framework: React (Latest Version)

- React sends and receives HTTP/HTTPS requests to and from Flask.
- Incoming data from Flask includes recipe recommendations and user profile details.
- Outgoing data to Flask consists of user inputs like registration details and preferences.

Services & Communications:

- Flask acts as the middleman, processing frontend requests, and sending appropriate responses back to the frontend after interacting with the MySQL database.
- Communications between the frontend and backend follow RESTful API standards.

Data Sharing Mechanism:

- JSON (JavaScript Object Notation) format is used for data exchange between React and Flask.

Implementation Constraints:

- ACID (Atomicity, Consistency, Isolation, Durability) properties will be maintained within MySQL.
- Global data areas should be minimized to prevent potential data inconsistencies.

3.3 Communications Interfaces

Web Browser Communications:

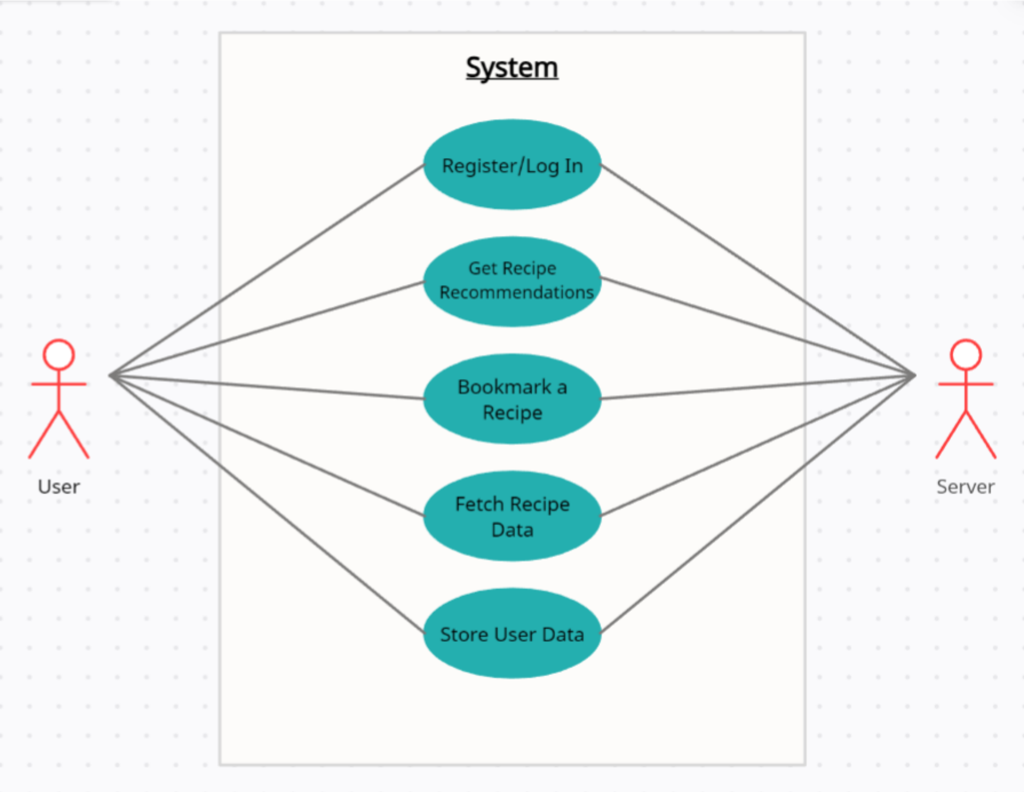
- Purpose: For users to interact with the Recipe Recommendation System.
- Protocol: HTTP/HTTPS for browser-server communication.
- Message Format: JSON for data exchange between frontend and backend.
- Security: Communications, especially user data and credentials, will be encrypted using HTTPS. SSL/TLS certificates will ensure data privacy.
- Data Transfer Rates: Approximate rate of 100-500 requests per second, depending on server scalability and user traffic.

Database Communications:

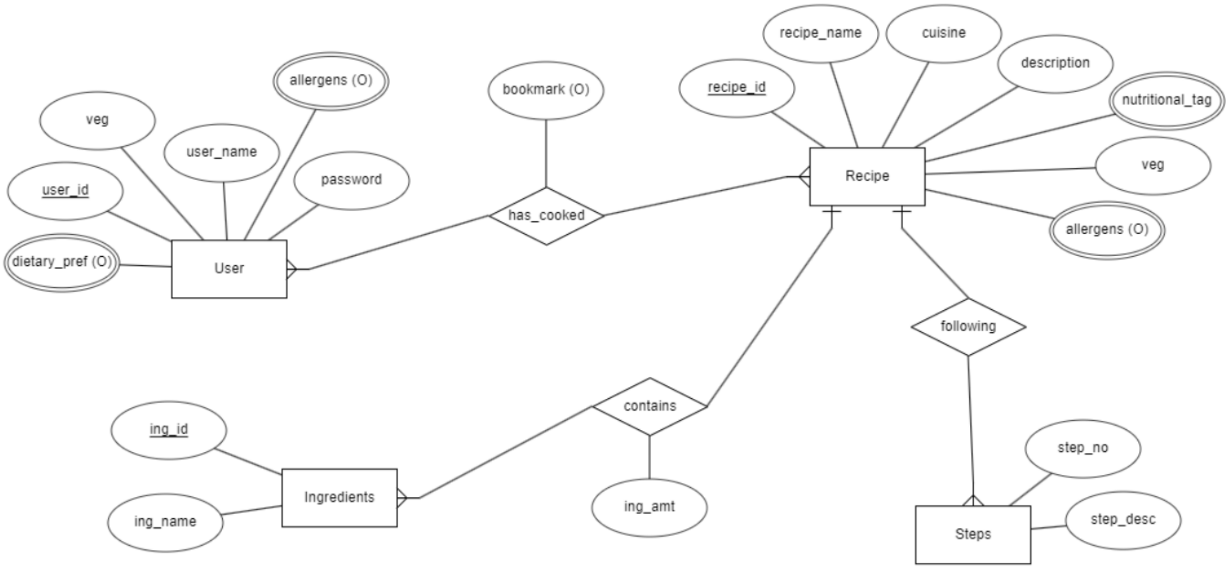
- Purpose: Fetching and storing user data, recipes, and other information.
- Protocol: MySQL's native protocol for interaction with the Flask backend.
- Message Format: SQL queries for data operations.
- Security: Connections use encrypted MySQL connections. Passwords and data will be hashed or encrypted before storage.
- Data Transfer Rates: Approximate rate of 50-200 queries per second, based on query efficiency and database optimization.

4. Analysis Models

Use Case Diagram:



E-R Diagram:



5. System Features

5.1. User Registration/Login

Description and Priority: High priority. Registered users can log in to the system.

Stimulus/Response Sequences:

- User initiates registration/login.
- User provides a valid username and password.
- System authenticates user information.
- System grants access to the user profile.

Functional Requirements:

- Authenticate user credentials.
- Create secure user sessions.
- Handle login errors (e.g., incorrect password, non-existent username) and provide informative error messages.

5.2. User Profile Management

Description and Priority: Medium priority. Users can create, edit, and update their profiles, including personal information and dietary preferences.

Stimulus/Response Sequences:

- User navigates to the profile page.
- Users update personal information or dietary preferences.
- System saves user profile changes.

Functional Requirements:

- Provide user-friendly profile management interfaces.
- Allow users to update profile information.
- Handle profile update errors (e.g., invalid data) and provide informative error messages.

5.3. Recipe Recommendation

Description and Priority: High priority. Users receive recipe recommendations based on their pantry contents, dietary preferences, and user history.

Stimulus/Response Sequences:

- Users set dietary preferences, and request recipe recommendations.
- System generates and displays recommended recipes.

Functional Requirements:

- Implement recommendation algorithms
- Consider user pantry contents, dietary preferences, and previous interactions.
- Generate recipe recommendations based on relevant factors.
- Provide clear and detailed recipe suggestions.
- Depending on the system's complexity and specific requirements, additional features and use cases may be necessary.

5.4. Bookmark Favorites

Description and Priority: Medium priority. Users can bookmark their favorite recipes for easy access.

Stimulus/Response Sequences:

- Users mark a recipe as a favorite.
- Users access their list of favorite recipes.

Functional Requirements:

- Implement a bookmarking feature.
- Allow users to add and remove recipes from their favorites.
- Provide a user-friendly interface to access bookmarked recipes.

5.5. Recipe Details

Description and Priority: High priority. Display detailed recipe information, including ingredients, step-by-step instructions, and nutritional information.

Stimulus/Response Sequences:

- User selects a recipe.
- System displays the recipe details.

Functional Requirements:

- Store comprehensive recipe data, including ingredients and instructions.
- Retrieve and display recipe details.
- Calculate and display nutritional information.

6. Other Nonfunctional Requirements

6.1. Performance Requirements

Response Time:

- Requirement: The system must respond to user interactions, such as recipe searches and profile updates, within 5 seconds.
- Rationale: Quick response times enhance the user experience, ensuring that users can interact with the system without delays.

Database Query Time:

- Requirement: Database queries, including user data and recipe retrieval, should execute within 1 second.
- Rationale: Efficient database access is critical for a responsive system.

Concurrent User Handling:

- Requirement: The system must support a minimum of 1,000 concurrent users without a significant degradation in performance.
- Rationale: Scalability is essential to accommodate a growing user base.

Image Loading Time:

- Requirement: Recipe images should load within 3 seconds on both web and mobile platforms.
- Rationale: Fast image loading enhances the visual appeal of the system.

Data Synchronization Time:

- Requirement: Data synchronization across different user devices should occur within 15 minutes of an update.
- Rationale: Timely data synchronization ensures a consistent user experience.

6.2. Safety Requirements

Allergen Warnings:

- Requirement: The system must provide allergen warnings for recipes, especially for users with specific food allergies or dietary restrictions.
- Safeguard: Include a feature that allows users to specify their food allergies and preferences, and display allergen warnings on recipes that may contain allergenic ingredients.

User Interaction Safety:

- Requirement: The user interface must be designed with user safety in mind. The system should not encourage unsafe cooking practices or promote risky behaviour.
- Safeguard: Ensure that recipe instructions are clear and safe, providing necessary safety tips, such as proper food handling, temperature control, and potential hazards (e.g., hot surfaces, sharp tools).

Avoiding Harmful Recommendations:

- Requirement: The system must not recommend recipes or ingredients that may be harmful to a user's health, taking into account dietary restrictions and health requirements.
- Prevention: Implement filters and validation mechanisms to exclude harmful or inappropriate ingredients or recipes based on user preferences and dietary restrictions.

Safety Education and Disclaimer:

- Requirement: Include a disclaimer stating that users are responsible for their safety in the kitchen and should follow local food safety regulations.
- Safeguard: Display safety disclaimers or tips in prominent areas, such as the initial app setup, recipe instructions, or user profile.

6.3. Security Requirements

Data Encryption:

- Requirement: User data, including personal information and preferences, must be encrypted both in transit and at rest.

User Authentication:

- Requirement: Implement secure user authentication, including strong password policies.

Access Control:

- Requirement: Enforce role-based access control to restrict unauthorized access to system functionalities.

Secure API Communication:

- Requirement: Ensure secure communication between the system and external data sources or APIs.

Secure Password Handling:

- Requirement: Store user passwords securely using salted and hashed methods.

Regular Security Audits:

- Requirement: Conduct regular security audits and vulnerability assessments to identify and address potential security issues.

Security Certifications:

- Requirement: Obtain relevant security certifications, if applicable, to demonstrate adherence to security standards.

User Data Deletion:

- Requirement: Allow users to request the deletion of their personal data, in compliance with data privacy regulations.

Logging and Monitoring:

- Requirement: Implement logging and monitoring of user activities for security and auditing purposes.

Incident Response Plan:

- Develop an incident response plan to address security breaches or data incidents promptly and effectively.

6.4. Software Quality Attributes

*Usability***:**

- Requirement: The user interface should be intuitive and user-friendly, with a low learning curve.
- Preference: High usability is a top priority, as it directly impacts user satisfaction.

*Reliability***:**

- Requirement: The system should be available and responsive, with minimal downtime.
- Preference: High reliability is crucial to prevent user frustration.

*Maintainability***:**

- Requirement: Code and database structures should be well-organized and documented for easy maintenance and updates.
- Preference: High maintainability ensures long-term system health.

*Security***:**

- Requirement: Robust security measures must protect user data and the system from threats.
- Preference: Security is a non-negotiable requirement.

*Performance***:**

- Requirement: The system should provide responsive recipe recommendations and quick access to features.
- Preference: High performance is crucial to enhance user experience.

Scalability:

- Requirement: The system should scale to accommodate increasing numbers of users and data.
- Preference: High scalability supports system growth.

Interoperability:

- Requirement: The system should be able to interact with external data sources, APIs, and devices.
- Preference: High interoperability allows for broader integration capabilities.

Adaptability:

- Requirement: The system should be adaptable to changing dietary trends, preferences, and culinary data sources.
- Preference: High adaptability keeps the system relevant over time.

Portability:

- Requirement: The system should work on multiple platforms, including web and mobile devices.
- Preference: High portability ensures accessibility for a broader user base.

Correctness:

- Requirement: The system should provide accurate recipe recommendations and nutritional information.
- Preference: High correctness is essential to fulfil the system's core purpose.

6.5. Business Rules

User Registration:

- Rule: Users must register an account to access personalized features.
- Domain Requirement: Collect user data for registration, including username, and password.

Login Authentication:

- Rule: Only registered users can log in to the system.
- Domain Requirement: Implement secure user authentication.

User Profiles:

- Rule: Users can create and update their profiles, including personal information, dietary preferences, and profile pictures.
- Domain Requirement: Create a user profile database.

Recipe Recommendations:

- Rule: Users receive recipe recommendations based on their pantry contents, dietary preferences, and user history.
- Domain Requirement: Implement recommendation algorithms.

Favourite Recipes:

- Rule: Users can save their favourite recipes for easy access.
- Domain Requirement: Implement a bookmarking feature.

Dietary Requirements:

- Rule: Users can specify dietary requirements, such as vegetarian, gluten-free.
- Domain Requirement: Store user dietary preferences.

Allergen Warnings:

- Rule: The system must provide allergen warnings for recipes that may contain allergenic ingredients based on user preferences.
- Domain Requirement: Develop allergen detection logic.

Recipe Nutritional Information:

- Rule: The system must provide nutritional information for recipes.
- Domain Requirement: Integrate nutritional data sources.

Appendix A: Glossary

Adaptability: The system's ability to adjust to changing dietary trends, preferences, and culinary data sources to remain relevant over time.

Allergen Warnings: Notifications provided by the system for recipes containing allergenic ingredients, especially catering to users with specific food allergies or dietary restrictions.

Allergy-Prone Individuals: People with specific dietary restrictions or allergies who often find it challenging to locate suitable recipes.

Aspiring Home Cooks: Individuals interested in expanding their culinary skills and looking for new recipes to try.

Avoiding Harmful Recommendations: The requirement that the system does not suggest recipes or ingredients that may be harmful to a user's health, considering dietary restrictions and health requirements.

Backend Framework: The software framework used to develop the backend of the system, which is Flask in this case.

Bookmark Favourites: The feature allows users to mark and save their preferred recipes for easy access.

Coexistence: The ability of the system to operate peacefully alongside other web applications without interfering with their functionality.

Concurrent User Handling: The system's capability to support a specified number of users simultaneously without a significant decrease in performance.

Correctness: The requirement that the system provides accurate recipe recommendations and nutritional information, ensuring the fulfilment of its core purpose.

Database: MySQL (Version 8.0): The database management system used in the system, which is MySQL in this case, specifying the version.

Database Communications: The processes and mechanisms related to the exchange of data between the system and the database.

Database Query Time: The requirement that database queries, including user data and recipe retrieval, should execute within a specified time frame.

Dataset Collection: The process of gathering a diverse and high-quality dataset of recipes with detailed information, including ingredients, instructions, and nutritional data.

Data Sharing Mechanism: The method or format used for data exchange between system components, often specified as JSON (JavaScript Object Notation) in this context.

Data Synchronisation Time: The requirement that data synchronisation across different user devices should occur within a specified time frame.

Deployment and Scalability: The system's capability to scale to handle a specified number of concurrent requests and the processes involved in deploying the system.

Description and Priority: The detailed explanation and importance level assigned to a specific feature or requirement within the system.

Developers: Individuals responsible for building and developing the system's software components and features.

Dietary Requirements: The dietary restrictions and preferences specified by users, such as vegetarian or gluten-free, that the system must accommodate in its recommendations and features.

Documentation Writers: Individuals responsible for creating comprehensive and accurate documentation for the software system.

Efficiency and Convenience: Characteristics ensuring the software performs its functions in a timely, effective manner and is user-friendly.

End Users: The individuals or entities who will interact with and utilise the software product upon its completion.

ER Diagram: (Entity-Relationship Diagram) - A visual representation of different entities within a system and their interrelationships.

External Libraries and Dependencies: Third-party codes, frameworks, or tools that the software relies upon to perform certain functions.

Frontend Framework: React (Latest Version): A JavaScript library utilised for developing the user interface of the application.

Functional Requirements: Descriptions of specific behaviours or functions of the system under defined conditions.

Gym Freaks: A user persona which may prioritise recipes and food recommendations conducive to a gym-friendly diet.

Hardware Platform: The physical technological environment (e.g., servers, computers, or mobile devices) where the software system will be deployed and operated.

Image Loading Time: The duration taken to successfully display an image on the user interface once requested.

Implementation Constraints: Limitations and restrictions that could affect the development and deployment of the software.

Incident Response Plan: A predetermined strategy or approach detailing the processes to follow when a cybersecurity incident occurs.

Integration and Compatibility: The software's ability to operate with and alongside other systems or platforms without conflict.

Interoperability: The capability of the software to exchange and utilise information across different systems and environments.

Landing Page: The initial web page users encounter, often serving as an introduction or overview of the application.

Language and Framework Limitations: The restrictions or shortcomings experienced due to the chosen programming languages and frameworks.

Logging and Monitoring: The practice of recording system activities and overseeing operational processes to ensure system health and security.

Login Authentication: The process of verifying a user's identity by validating their credentials upon login.

Login Page: A web page where users provide credentials (e.g., username, password) to access personalised or restricted content.

Maintainability: The ease with which the software system can be modified to change or add functionalities, fix issues, and improve performance.

Maintenance and Updates: Ongoing activities to ensure the software continues to perform as required, including modifications, corrections, and enhancements.

Marketing Staff: Team members responsible for promoting the recipe recommendation system to users and stakeholders.

Memory and Storage: Pertaining to the hardware and software capabilities that manage and store data for the system.

Network: The infrastructure that facilitates the data exchange and communication between different parts of the system.

Operating System: The software that supports the system's basic functions, such as scheduling tasks and controlling peripherals.

Password Handling: Procedures and protocols employed to securely manage user password creation, storage, and verification.

Performance: The efficiency and capability of the system in processing requests and delivering services.

Personalization: Customising user experiences and recommendations based on their preferences and historical data.

Portability: The system's ability to operate and adapt across various environments and platforms without extensive reconfiguration.

Profile Page: A user interface that displays and allows the user to interact with their personal data and preferences.

Project Managers: Individuals tasked with planning, executing, and closing the project, ensuring it meets its goals and success criteria.

Recipe Details: Specific information about a recipe, such as ingredients, instructions, and possible variations.

Recipe Nutritional Information: Data relating to the nutritional content of a recipe, including calories, macronutrients, and micronutrients.

Recipe Page: A user interface that presents detailed information and instructions for a specific recipe.

Recipe Recommendations: Suggested recipes provided to the user based on various criteria, such as ingredients available and preferences.

Recommendation System: A subsystem or algorithm designed to provide personalised content suggestions to users.

Reliability: The system's ability to consistently perform its intended functions under specified conditions.

Response Time: The duration between a user's action/request and the system's reaction/response.

Safety Education and Disclaimer: Information and warnings provided to users about safe cooking practices and a disclaimer regarding dietary advice.

Scalability: The system's capacity to handle growth in users and data without compromising performance.

Secure API Communication: The safeguarded and encrypted exchange of data and information between the system and external data sources or APIs, ensuring data privacy and integrity (Reference: Glossary - Services & Communications).

Security: The protection of the system's data and user information from unauthorised access, breaches, and threats, encompassing measures to maintain data confidentiality, integrity, and availability (Reference: Glossary - Security Requirements).

Security Considerations: The set of requirements and strategies to ensure the safety and integrity of the system, including measures to safeguard against potential threats and vulnerabilities (Reference: Glossary - Security Requirements).

Services & Communications: The components and processes related to data exchange and interactions within the system, encompassing communication protocols and data transfer mechanisms (Reference: Glossary - Services & Communications).

Stimulus/Response Sequences: The series of actions or events initiated by the user (stimulus) and the corresponding system reactions (responses) that describe how the system should behave in various scenarios (Reference: Glossary - Stimulus/Response Sequences).

Testers: Individuals responsible for creating test cases and scenarios to validate that the system meets its requirements and functions as intended.

Usability: The degree to which the system's user interfaces are intuitive and user-friendly, ensuring a low learning curve and high user satisfaction.

Use Case Diagram: A visual representation that illustrates the interactions between system components and external actors, helping to understand and document system functionality.

User Authentication: The process of verifying the identity of users accessing the system, typically involving secure login procedures.

User History: A record of a user's interactions and activities within the system, often used for personalization and tracking purposes.

User Interaction Safety: Design considerations to ensure that the user interface does not encourage unsafe cooking practices or risky behaviour.

User Profile Management: Functionality allowing users to create, edit, and update their profiles, including personal information and preferences.

User Profiles: Individual user accounts with personal information and preferences that the system uses to customize the user experience.

User Registration/Login: The combination of user registration (account creation) and user login (access to the system with credentials).

Web Browser Communications: The means by which users interact with the system using web browsers, often involving HTTP/HTTPS protocols for communication.

Web Browsers: Software applications used to access and interact with web-based systems, including modern web browsers like Google Chrome and Mozilla Firefox.

Appendix B: Field Layouts

Information required to register/login the end user

Field	**Length(bytes)**	**Data type**	**Description**	**Is Mandatory**
Username	20	Alphanumeric		Y
Password	20	Alphanumeric		Y
Customer Name	20	String		Y
Veg	1	Boolean	Veg/Non-veg	Y
Allergens	45	String	If any relevant allergies	N
Dietary Preference	45	String	If customer has dietary preference	N

Appendix C: Requirement Traceability Matrix

Sl. no.	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID